



Umsetzung eines alternativen Benutzerinterfaces für Smart Homes

Bachelorarbeit

von

Florian Fiebig

Matrikelnummer: 1356967

An der Fakultät für Wirtschaftswissenschaften
Institut für Angewandte Informatik und
Formale Beschreibungsverfahren

Gutachter:	Prof. Dr. Hartmut Schneck
Betreuender Mitarbeiter:	Daniel Pathmaperuma

Eidesstattliche Erklärung

Ich versichere hiermit wahrheitsgemäß, die Arbeit und alle Teile daraus selbständig oder in Zusammenarbeit mit meinem Team angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderung entnommen wurde.

Karlsruhe, den 30.04.2013

Florian Fiebig

Abstract

Diese Arbeit befasst sich mit dem Design und der Implementierung eines alternativen Userinterfaces für Smart Homes auf Basis der Open-Source Prototyping-Plattform Arduino.

Durch den Wandel am Energiemarkt in Form eines stetig wachsenden Anteils der erneuerbaren Energien und die damit verbundene zunehmend dezentrale Einspeisung entstehen Probleme wie starke Fluktuationen der Energiemenge. Dies führt zu einer erhöhten Belastung der Netze und macht eine Abstimmung der Energienachfrage mit dem Energieangebots nötig. In Smart Home Haushalten können Verbraucher wie Waschmaschine oder Trockner so gesteuert werden, dass sie Strom möglichst zu dem Zeitpunkt verbrauchen, zu dem er günstig ist oder wenn gerade ein großer Anteil aus der eigenen Photovoltaik-Anlage bzw. BHKW verfügbar ist. Jedoch ist die Steuerung, genauer die Benutzerschnittstellen, der am Markt verfügbaren Systeme entweder zu teuer oder ihre Benutzung ist für Personen mit einer Technikaversion zu kompliziert.

Im Rahmen dieser Arbeit wurde daher ein Prototyp eines alternativen Userinterfaces für Smart Homes mit der folgenden Funktionalität designet und entwickelt. Mit Hilfe von an anderer Stelle prognostizierten Daten, dem Gütemaßes des verfügbaren Stroms, kann der Beendigungszeitpunkt für Verbraucher wie Spülmaschine, Trockner oder Waschmaschine intelligent gesteuert werden. Dabei wurde ein Hauptaugenmerk auf eine intuitive Bedienung und niedrige Kosten für den Bau des Interfaces gelegt. Um diese Ziele zu erreichen, wurden bei der Designidee, sowie bei der anschließenden Umsetzung zum einen verschiedene Empfehlungen von Normen aus dem Bereich der Mensch-System-Interaktion, und zum anderen bekannte User-Design-Guidelines zur Hilfe gezogen. Der als Ergebnis der Arbeit entstandene Prototyp verfügt bewusst über nur eine Eingabemöglichkeit und gibt dem potentiellen Benutzer zu jeder Zeit ein Feedback darüber, welche Aktion er gerade ausführt. Dies sollte zu einer intuitiven Benutzungs-Erfahrung führen.

Inhaltsverzeichnis

Abstract	iii
1 Einleitung	1
1.1 Problemstellung & Zielsetzung	1
1.2 Gliederung	3
2 Grundlagen	5
2.1 Energiewende	7
2.2 Benutzerschnittstelle	8
2.3 Smart Homes	9
2.4 Userinterfaces auf dem Smart Home Markt	9
3 Design	15
3.1 Richtlinien der Mensch-Computer-Interaktion	15
3.2 Userinterface Design Guidelines	17
3.3 Designidee	18
4 Entwicklung	21
4.1 Arduino-Framework	21
4.2 Komponenten	21
4.3 Umsetzung	27
5 Evaluation	31
6 Zusammenfassung & Ausblick	33
6.1 Fazit	33
6.2 Ausblick	33

Abkürzungsverzeichnis

EEG Erneuerbare-Energien-Gesetzes

IKT Informations- und Kommunikationstechnologien

DSRM Design Science Research Methodology

DS Design Science

DMS Demand Side Management

MMS Mensch-Maschine-Schnittstelle

MCI Mensch-Computer-Interaktion

MSI Mensch-System-Interaktion

EMS Energy Management System

EMP Energy Management Panel

UDG User Design Guidelines

USB Universal Serial Bus

TKG Telekommunikationsgesetz

BNA Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen

SRD Short-Range-Devices

ISM Industrial, Scientific and Medical Band

Abbildungsverzeichnis

1	Design Science Research Methodology (DSRM) (aus [PTRC07])	6
2	Fernbedienung zur Steuerung des RWE SmartHomes	10
3	Verschiedene Ansichten der Benutzeroberfläche der EVN SmartHome Ap- plikation für Android	11
4	Übersicht der Benutzeroberfläche des nest Thermostats aus [Nes14]	12
5	Übersicht der Benutzeroberfläche des EMP aus [BKS12]	13
6	Die sich am meisten unterscheidenden Farben aus [Joh10]	19
7	Erster Prototyp aus Pappe	20
8	Obere Seite des Arduino Uno R3	22
9	Im Prototyp verwendete Ein- und Ausgabekomponenten	23

1 Einleitung

Durch den Wandel am Energiemarkt nehmen regenerative Energiequellen einen immer höheren Stellenwert ein. In Deutschland betrug im Jahr 2013 der Anteil erneuerbarer Energien am Bruttostromverbrauch, laut der Arbeitsgemeinschaft Energiebilanzen e.V., 24.7% [AGE13]. In der Novellierung des Erneuerbare-Energien-Gesetzes (EEG) von 2012 hat sich die Bundesregierung das Ziel gesetzt, bis zum Jahr 2050 80 % des Bruttostromverbrauchs aus erneuerbaren Energien zu beziehen [BMW13]. In Folge dieser eigentlich positiven Änderung der Anteile im Energiemix entstehen allerdings auch Nachteile. Durch den erhöhten Einsatz von regenerativen Energiequellen und der damit einhergehenden dezentralen und volatilen Einspeisung ergibt sich, da diese Veränderung unter anderem in Fluktuationen im Energienetz resultiert, ein erhöhter Regelenenergiebedarf [Pic09]. Ein weiteres Problem, besonders bei Windenergie, ist die Tatsache, dass der Hauptteil in Norddeutschland erzeugt wird und anschließend zu den Großverbrauchern in den Süden und Westen Deutschlands transportiert werden muss [BMW13]. Solche zunehmenden Anforderungen an das Stromnetz führen zu einer steigenden Komplexität und einem erhöhten Anspruch an die Netze. Um einerseits das Ziel der Energiewende zu erreichen, und andererseits dessen Nachteile zu beseitigen müssen neue Wege und Mittel gefunden werden.

Auf Grund der Tatsache, dass der Anteil von Strom aus regenerativen Energiequellen fortwährend weiter steigt, kommt es zu den im letzten Abschnitt angesprochenen Problemen. Um diese zu bewältigen ist unter anderem eine Abstimmung der Energienachfrage mit dem Energieangebot nötig. Dazu können neue Informations- und Kommunikationstechnologien (IKT) eingesetzt werden. Intelligente Häuser und Wohnungen, sogenannte Smart Homes, setzen auf genau solche Technologien und können bei der Regulation von Angebot und Nachfrage des Stroms helfen. In einem Haushalt ist die Zeit, zu der Großverbraucher wie Trockner oder Spülmaschine aktiv sein sollen, nicht fest und somit in gewissen Grenzen planbar. Somit ist es bis zu einem gewissen Grad möglich Fluktuationen des Stromangebots durch regulierbare Energieverbraucher und -Quellen besser in Einklang zu bringen.

1.1 Problemstellung & Zielsetzung

Essentiell ist für Smart Homes, neben ihrer eigentlichen Funktionalität, die Stelle an dem der Nutzer mit dem dahinter liegenden System in Verbindung kommt. Diese Mensch-Computer-Schnittstellen sind in den etablierten Systemen meist als mobile Applikation realisiert und können dem Nutzer zahlreiche Informationen über die Energieverbraucher und -Erzeuger des Haushalts anzeigen. Aus der Wirtschaft, sowie aus wissenschaftlichen Projekten existiert eine große Anzahl verschiedener Benutzerinterfaces für Smart Homes.

Jedoch sind besonders Lösungen aus dem wissenschaftlichen Bereich oftmals sehr technisch und daher für Anwender, die wenige Kenntnisse über Smart Homes besitzen oder eher technikavers sind, nur schwer zu handhaben. Produkte aus der Wirtschaft sind zwar vereinzelt einfacher zu bedienen, gehen jedoch mit hohen Beschaffungs- und Einrichtungskosten einher. Zudem sind bisher (Stand März 2014) keine Produkte, die dem Endverbraucher Anreize setzen ein Gerät zu einem bestimmten Zeitpunkt laufen zu lassen und somit auf die variable verfügbare Strommenge zu reagieren, auf dem Markt verfügbar.

Im Rahmen dieser Arbeit soll daher mit der Problematik, wie sich ein alternatives Benutzerinterface für Smart Homes entwerfen und umsetzen lässt, beschäftigt werden. Realisiert werden soll das Projekt, nach Vorgabe des Betreuers, auf Basis der Open Source Elektronik Prototyping-Plattform Arduino [Ard14].

Folgende Ziele sollen dabei erreicht werden:

- Im Gegensatz zu bereits am Markt erhältlichen Produkten soll die Benutzung deutlich vereinfacht werden. Daher soll das Interface so konstruiert sein, dass selbst für technikaverse Personen eine Bedienung des Gerätes ohne Einweisung möglich ist.
- Der User soll bei Benutzung des Interfaces jederzeit ein direktes Feedback über den Zustand des Systems erhalten. Dies kann entweder über akustische oder optische Signale geschehen.
- Daten, die innerhalb eines Smart Homes zwischen den verschiedenen Aktoren übertragen werden, sind datenschutztechnisch betrachtet hochsensible Daten. Daher soll das hier umzusetzende System auf die Nutzung von Cloud-basierter Services verzichten. Weiterhin muss durch den Einsatz von Verschlüsselung sichergestellt werden, dass Dritte keinen Zugang zur Smart Home Steuerung erlangen können.
- Derzeit werden im *House of Living Labs* des *Forschungszentrum für Informatik* im Modell Smart Home *FZI Living Lab Smart Energy* für die kabellose Kommunikation ZigBee-Module eingesetzt. Diese nutzen das 2,4 GHz ISM-Band. Da weitverbreitete Standards wie Bluetooth oder WLAN auch auf 2,4 GHz arbeiten, ist dieses Frequenzband oft überlastet. Diese Überlastung führt zu Interferenzen. Die drahtlose Verbindung soll daher so gestaltet werden, dass im Gegensatz zum jetzigen Setup die Reichweite erhöht und Interferenzen vermieden werden.
- Es soll nur geringe Anforderungen für die Teilnahme an der Nutzung geben. Das bedeutet vor allem, dass der Preis des Interfaces möglichst gering ausfallen soll. Daher sollen alle Komponenten zusammen einen Gesamtpreis von weniger als 100 € haben.
- Im späteren Verlauf soll das hier umzusetzende Interface in folgendem Szenario eingesetzt werden können. Eine zentrale Einheit des Smart Homes, die für die eigent-

liche Steuerung der intelligenten Geräte zuständig ist, prognostiziert Daten über die Verfügbarkeit und den Preis des im Smart Home verfügbaren Stroms für die nächsten 24 Stunden. In der vorliegenden Arbeit wird angenommen, dass es sich bei dieser Einheit um einen Raspberry Pi handelt. Die vorhergesagten Daten werden im weiteren Verlauf als Gütemaß bezeichnet und sollen dem hier zu entwickelnden Gerät übermittelt werden. Unter der Berücksichtigung des Gütemaßes soll das Userinterface den Benutzer bei der Wahl eines *intelligenten* Beendigungszeitpunkts der Großverbraucher unterstützen.

Als Ausgangspunkt wird die Bachelorarbeit von Melda Henden [Hen13] verwendet. In dieser werden zuerst eine Auswahl der am Markt befindlichen Smart Home Interfaces vorgestellt und anschließend mehrere Vorschläge zu alternativen Benutzerinterfaces entwickelt. Eine dieser Möglichkeiten ist ein intelligenter Küchenwecker namens *SmartEi*, mit dem der Nutzer eines Smart Homes Haushaltsgeräte wie Ofen, Kühlschrank oder Waschmaschine steuern kann.

1.2 Gliederung

Eine Orientierung für den Aufbau dieser Arbeit liefert im wesentlichen die DSRM von Peffers et al. [PTRC07]. Dieses besteht im Wesentlichen aus einem Prozessmodell mit sechs verschiedenen Aktivitäten, welche im Anschluss an diesen Abschnitt, zu Beginn von Kapitel 2, vorgestellt werden. Die *Identifizierung des Problems und die Motivation* wurden bereits behandelt und finden ihren entsprechenden Teil am Anfang des Kapitels 1. Anknüpfend wird in 1.1 näher auf die Problematik eingegangen und im Anschluss Ziele für eine mögliche Lösung Problems definiert.

Im Anschluss daran werden in Kapitel 2 zunächst Grundlagen rund um Smart Homes und der Mensch-Computer-Interaktion vermittelt sowie existierende Userinterfaces für Smart Homes vorgestellt.

Anschließend werden in Kapitel 3 die Grundlagen für das Design des alternativen Userinterfaces erläutert und darauf aufbauend das Gestaltung des ersten Prototyps vorgestellt. Im weiteren Verlauf des Kapitels wird mit Hilfe von Normen und Userinterface Design Guidelines eine Grundlage für die Entwicklung der alternativen Mensch-Maschinen-Schnittstelle, dessen erster Entwurf am Ende des Kapitels gezeigt wird, gegeben.

In Kapitel 4 wird zuerst ein Überblick über das Arduino Ökosystem gegeben. Folgend wird die Komponentenauswahl für das alternative Userinterface dargelegt und begründet. Im letzten Teil der Passage wird die grundlegende Programmstruktur von Arduino Programmen erläutert und anschließend die softwareseitige Umsetzung detailliert erklärt. Kapitel 3 und 4 entsprechen somit Aktivität 3, der *Design & Entwicklung des Artefakts*.

Die *Demonstration* und *Evaluation*, respektive Aktivität 4 und 5 finden sich im 5. Ka-

pitel wider. Zum Schluss werden in Kapitel 6, welches in etwa der sechsten Aktivität gleichkommt, die Ergebnisse der Arbeit zusammengefasst und ein Ausblick auf mögliche zukünftige Arbeiten gegeben.

2 Grundlagen

In diesem Kapitel wird zuerst mit Hilfe einer Methodik aus der *Design Science* (DS) die Herangehensweise an die Bearbeitung dieser Arbeit dargelegt. Anschließend werden die Zusammenhänge zwischen den Begriffen Energiewende, E-Energy sowie Smart Homes und dieser Arbeit aufgezeigt.

Um den Vorgang der Entwicklung und der Umsetzung des alternativen Benutzerinterfaces nachvollziehbar und systematisch zu gestalten, wird zu diesem Zweck ein Ansatz aus der *Design Science* (DS) verwendet. Der Begriff der DS wurde 1965 von R. Buckminster Fuller [Ful65] eingeführt und als systematische Form des Gestaltens definiert. Die Abbildung 1 zeigt die von Peffers et al. [PTRC07] entwickelte Design Science Research Methodology (DSRM). Die Methodik wurde unter der Berücksichtigung von drei Zielen entwickelt: Sie soll konsistent mit vorangegangener Literatur sein, ein nominales Prozessmodell zur Ausübung von Design Science (DS), sowie ein mentales Modell für das Präsentieren und Evaluieren von Forschung im Bereich der DS sein. Das Prozessmodell setzt sich aus den sechs nachfolgenden Aktivitäten zusammen: Identifizierung des Problems und Motivation, Definieren der Ziele für eine Lösung, Design & Entwicklung des Gegenstands, Demonstration, Evaluation und Kommunikation. Im Folgenden Teil werden zunächst die einzelnen Bausteine des Prozessmodells näher vorgestellt. Die Zuordnung der jeweiligen Aktivitäten wurde bereits in Kapitel 1 in Abschnitt 1.2 vorangestellt. Da sich diese Arbeit im Wesentlichen mit Entwicklung und Umsetzung eines alternativen Interfaces beschäftigt, wird der Fokus auf die Aktivität *Design & Entwicklung* und somit auf Kapitel 3 und 4 gelegt.

Aktivität 1 - Identifizierung des Problems und Motivation:

Zunächst wird das spezifische Problem definiert. So ist es möglich später ein Artefakt zu entwickeln, welches eine entsprechend gute Lösung bietet. Zudem soll der Wert einer Lösung für das Problem aufgezeigt und begründet werden.

Aktivität 2 - Definieren der Ziele für eine Lösung:

Durch die in Aktivität 1 festgesetzte Problembeschreibung werden nun die Ziele für eine Lösung definiert. Diese werden aus der Problemstellung abgeleitet und können dabei entweder qualitativ oder quantitativer Natur sein.

Aktivität 3 - Design & Entwicklung des Artefakts:

Peffers [PTRC07] fasst den Begriff des Artefakts, welches in diesem Schritt erstellt wird, sehr weit. Das Artefakt muss kein physischer Gegenstand wie ein Prototyp, sondern kann auch ein Konstrukt, ein Modell oder eine Methode sein.

Aktivität 4 - Demonstration:

Hier wird das Artefakt benutzt um das in Aktivität 1 beschriebene Problem zu lösen. Je nach Art und Form des Artefakts kann durch Simulationen, eine Fallstudie, einen Beweis oder eine andere adäquate Aktivität geschehen.

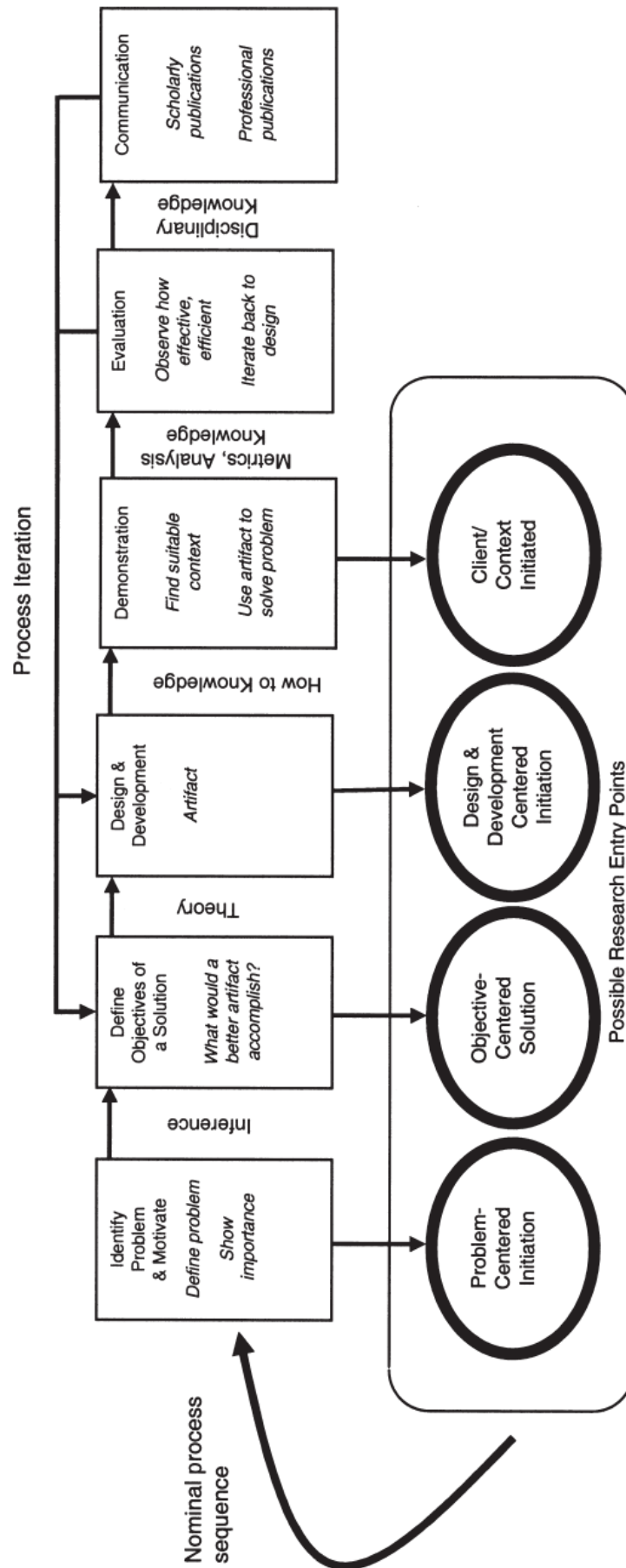


Abbildung 1: Design Science Research Methodology (DSRM) (aus [PTRC07])

Aktivität 5 - Evaluation:

In diesem Schritt werden die Ergebnisse aus der Demonstration mit Hinblick auf die gesteckten Ziele beurteilt. Es wird beurteilt an welchen Stellen die entwickelte Lösung den in *Definieren der Ziele für eine Lösung* Ansprüchen gerecht werden kann und an welcher Stelle nicht. Je nach Ergebnis kann es statt zum nächsten Schritt weiter zu gehen sinnvoll sein, zu Aktivität 2 bzw. Aktivität 3 zu springen um eine Iteration der Schritte durchzuführen.

Aktivität 6 - Kommunikation:

Im letzten Schritt werden das Problem und dessen Wichtigkeit sowie das entwickelte Artefakt und die Ergebnisse an relevante Zielgruppen kommuniziert.

2.1 Energiewende

In diesem Abschnitt wird der Begriff der Energiewende näher beleuchtet sowie die Hintergründe für eben diese aufgezeigt.

Die Zergliederung vertikal integrierter¹ Elektrizitätsversorgungsunternehmen sowie der Umstand, dass eine hohe Anzahl neuer Akteure in den Energiemarkt eindringen führen dazu, dass die Komplexität der Marktsituation merklich erhöht wird [Pic09]. Nicht zuletzt durch die Förderung der Bundesregierung für die Erzeugung erneuerbarer Energien im Sinne des EEG nimmt der Anteil an erneuerbaren Energien weiter zu und erhöht damit die Dezentralisierung der Energieerzeugung. Durch die Abhängigkeit der regenerativen Stromerzeuger vom Wetter kommt es zu erheblichen Schwankungen in der Strommenge und führt so zunehmend zu einer stärkeren Belastung der Netze. Eine Umgestaltung und Optimierung der Energieversorgung ist daher ein zentraler Punkt für die Energiewende [Pic09].

Um eine Lösung für die im vorigen Absatz angesprochenen Probleme zu finden ist ein erhöhter Einsatz von IKT sinnvoll. In Anbetracht dieser Tatsachen hat das *Bundesministerium für Wirtschaft und Technologie* in Kooperation mit dem *Bundesministerium für Umwelt, Naturschutz und Reaktorsicherheit* (BMU) im Jahr 2008 das Förderprogramm *E-Energy* ins Leben gerufen. *E-Energy* steht dabei für *Electronic Energy* bzw. dem Sinne nach für *Internet der Energie*. Ziel des Programmes ist es durch intelligenten Einsatz von IKT die Probleme des heutigen Energiemarktes zu lösen. Als Beispiele für solch einen Einsatz werden *virtuelle Kraftwerke*, *Smart Grids*, *Demand Side Management* (DMS) oder auch *Elektromobilität* genannt [Pic09].

¹Ein vertikal integriertes Energieversorgungsunternehmen ist nach Legaldefinition des §3 Nr.38 EnWG ein Unternehmen oder eine Gruppe von Unternehmen, „...wobei das betreffende Unternehmen oder die betreffende Gruppe in der Europäischen Union im Elektrizitätsbereich mindestens eine der Funktionen Übertragung oder Verteilung und mindestens eine der Funktionen Erzeugung oder Vertrieb von Elektrizität...“ [EnW14] wahrnimmt

Im September 2010 veröffentlichte das Bundeskabinett das *Energiekonzept für eine umweltschonende, zuverlässige und bezahlbare Energieversorgung* [Bun10]. Im Jahr 2011 hat die Bundesregierung auf Basis dieses Energiekonzepts, unter anderem durch die Festlegung auf quantitative Ziele, den Plan für einen umfassenden Umbau der Energienetzes in Deutschland mit stärkerer Ausrichtung auf regenerative Energien vorgelegt. Eines der quantitativen Ziele ist die Steigerung des Anteils erneuerbarer Energien am Bruttostromverbrauch bis zum Jahr 2050 auf mindestens 80 Prozent. Im Jahr 2013 betrug in Deutschland der Anteil erneuerbarer Energien am Bruttostromverbrauch 24.7% und war damit, relativ zu dem Vorjahreswert von 23,6%, um über 4,6% höher als 2012 [AGE13].

Auf diese Veränderungen reagierte die Bundesregierung unter anderem mit den im letzten Abschnitt aufgeführten Maßnahmen. Das vorig angesprochene Demand Side Management (DMS) oder auf Deutsch *Laststeuerung* bezeichnet im eigentlichen Sinne Maßnahmen der Energieversorgungsunternehmen, die den Verbrauchslastgang beeinflussen [vRGH10]. In diesem Zusammenhang ist die Steuerung des Energieverbrauchs mit Hilfe intelligenter Haushaltsgeräte ein integraler Bestandteil von Smart Homes und wird ebenfalls als DMS aufgeführt [Pic09]. DMS bezeichnet somit nicht nur ausschließlich Maßnahmen der Energieversorger. Diese Steuerung kann über verschiedenste Wege stattfinden, es findet jedoch immer eine Kommunikation mit dem Benutzer über irgend eine Art von Benutzerschnittstelle statt. Im nächsten Teilabschnitt soll daher zuerst der Begriff der Benutzerschnittstelle erklärt und im Anschluss daran der des Smart Homes genauer betrachtet werden.

2.2 Benutzerschnittstelle

Im Rahmen dieser Arbeit soll ein Bedienelement für Smart Homes entwickelt und umgesetzt werden. Benutzerschnittstellen sind der Punkt an dem ein Mensch mit einer Maschine in Verbindung tritt. Daher werden sie auch als Mensch-Maschine-Schnittstelle (MMS) bezeichnet. Es existieren außerdem weitere Bezeichnungen für den Begriff des Userinterfaces, daher werden im Kontext dieser Arbeit die Begriffe *Benutzerschnittstelle*, *Benutzungsschnittstelle*, *Mensch-Maschine-Schnittstelle* (MMS) und *Userinterface* substituierend eingesetzt. In diesem Abschnitt wird folgend das, was unter diesen Begriffen verstanden wird, näher beleuchtet.

Die Forschungsgebiet der Mensch-Computer-Interaktion (MCI) ist ein Teilgebiet der Mensch-System-Interaktion (MSI) und setzt sich aus vielen verschiedenen Fachrichtungen zusammen. Neben Informatik sind dies Psychologie, Arbeitswissenschaft, Ingenieurwissenschaften und Design-Disziplinen [MCI14]. Fallmann [Fal03] geht davon aus, dass MCI als Design-Orientiertes Forschungsfeld betrachtet werden muss, da das Design von Informations-, Interaktions- und Kommunikationstechnologie zentral für dieses ist. Aus

diesem Grund soll dem Design der hier entworfenen Mensch-Maschinen-Schnittstelle ein eigenes Kapitel gewidmet werden.

Eine Definition für Benutzungsschnittstelle findet sich in der in Abschnitt 3.1.2 beschriebenen Teilnorm 116 der Richtlinien der Mensch-System-Interaktion und lautet: „Die Benutzungsschnittstelle umfasst alle Bestandteile eines interaktiven Systems (Software oder Hardware), die Informationen und Steuerelemente zur Verfügung stellen, die für den Benutzer notwendig sind, um eine bestimmte Arbeitsaufgabe mit dem interaktiven System zu erledigen [DIN06].“ Beispielsweise können Userinterfaces einfache Dinge wie ein Lichtschalter oder auch kompliziertere wie die Kommandozeilenoberfläche eines Betriebssystems sein.

2.3 Smart Homes

In der Literatur gibt es viele verschiedene Bezeichnung für den Einsatz intelligenter Verbraucher in Wohnungen bzw. Häusern. Neben *Smart Home* werden viele bedeutungsähnliche bzw. bedeutungsgleiche Begriffe wie *Home Automation*, *Smart House*, oder auch *vernetztes Haus* verwendet. In nächsten Absatz wird genauer geklärt, was einen normalen Haushalt zu einem intelligente Haushalt bzw. *Smart Home* macht. Im danach folgenden Abschnitt werden verschiedene Produkte auf den Hauptaugenmerk ihres Aufgabengebietes untersucht.

Der Begriff des Smart Homes wird nicht einheitlich verwendet und in der Fachliteratur keine einheitliche Begriffsdefinition für *Smart Home* vorhanden. Erstmals wurde der Begriff von der *American Association of House Builders* im Jahr 1984 genannt. Heutzutage bezeichnet der Ausdruck *Smart Home* vor allen den Einsatz von IKT in Haushalten.

Eine einfache jedoch anerkannte Definition, wurde 2003 durch Intertek, ein Unternehmen für Qualitäts- und Sicherheitsdienstleistungen, im Rahmes des Projekts *DTI Smart Homes Project* veröffentlicht. Laut Intertek's Ansicht ist ein *Smart Home* „A dwelling incorporating a communication network that connects the key electrical appliances and services, and allows them to be remotely controlled, monitored or accessed.“ Es sind folglich drei Elemente [JyLY04], die einen herkömmlichen Haushalt *smart* machen. Diese sind das interne, kabel- oder kabellose, Netzwerk, ein intelligentes Bedienungselement in Form einer Schnittstelle zum System, sowie die automatisierte Nutzung von Geräten innerhalb des Haushalts.

2.4 Userinterfaces auf dem Smart Home Markt

Auf dem Markt für Smart Homes existieren viele unterschiedliche Ansätze für Benutzeroberflächen, wobei bezüglich ihres Anwendungsbereiches jeweils ein unterschiedlicher

Fokus gelegt wird. So gibt es auf der einen Seite Systeme, die eine neben der Steuerung von Geräten zusätzliche Informationen über das Smart Home anzeigen, auf der anderen Seite gibt es Systeme, die lediglich Verbraucher Ein- oder Ausschalten können. Bei jedem der Systeme interagiert der Endnutzer in verschiedenartiger Weise mit dem Smart Home, da jede Lösung ihr eigenes Userinterface mit sich bringt. Was Userinterfaces genau sind, wurde bereits in Abschnitt 2.2 erläutert, Ansprüche an diese, werden nicht an dieser Stelle sondern später im Kapitel in Abschnitt 3.1.2 geklärt.

Im Folgenden werden die auf dem Markt existierende Produkte bzw. Lösungen vorgestellt, wobei zuerst Beispiele aus der Wirtschaft und anschließend aus der Forschung gezeigt werden. Es wird jeweils das Produkt vorgestellt und anschließend neben der Funktionalität der Benutzerschnittstelle auch der Fokus der Lösung betrachtet.

2.4.1 Fernbedienung von RWE SmartHome

Das eigentliche Userinterface des RWE SmartHomes [RWE14a], welches über einen Browser genutzt werden kann, wurde in der Bachelorarbeit von Melda Henden [Hen13] ausführlich beschrieben und in seiner Funktionsweise erläutert. Da das im Rahmen dieser Arbeit entwickelte Interface in der Funktionsweise einer Fernbedienung ähnelt, wird es an dieser Stelle als zielführender angesehen die in Abbildung 2 gezeigte Fernbedienung des RWE SmartHomes [RWE14b] zu betrachten.



Abbildung 2: Fernbedienung zur Steuerung des RWE SmartHomes

Die Fernbedienung verfügt im unteren Bereich über eine Status LED und im oberen Bereich über acht Tasten. Jeder einzelnen Taste können ein oder mehrere Geräte des RWE SmartHomes zugeordnet werden und durch Druck auf die Taste ein bzw. ausgeschaltet werden. Damit können das Licht, die Heizung und andere elektronische Geräte ein- und ausgeschaltet werden. Weiterhin gibt RWE die Funkreichweite des auf 868,3 MHz senden-

den Gerätes mit 100 Metern im Freifeld an. Die im RWE SmartHome integrierten Geräte können somit nicht nur über die Benutzeroberfläche im Browser, sondern auch mit der Fernbedienung gesteuert werden. Eine Rückmeldung darüber, ob die gewünschte Funktion durch Druck einer Taste erfolgreich ausgeführt wurde, ist direkt nicht vorhanden. Jedoch wird eine fehlgeschlagene Übertragung durch kurzes dreimaliges Blinken der oben erwähnten Status LED signalisiert.

2.4.2 EVN Smart Home

Das EVN SmartHome ermöglicht eine Steuerung elektrischer Geräte und der Heizung über das Internet. Für die Regelung Heizkörper muss das herkömmliche Thermostat gegen das *EVN SmartHome Heizkörperthermostat* getauscht werden. Für die Fernsteuerung der elektrischen Geräte ist pro Gerät der *EVN SmartHome Zwischenstecker*, eine intelligente Steckdose, notwendig [EVN14]. So ist es theoretisch zwar möglich, alle sich im Haushalt befindlichen Geräte zu vernetzen, jedoch ist dies zum einen sehr teuer, zum anderen beschränkt sich die Fernsteuerung auf das Ein- und Ausschalten der Verbraucher. Als Schnittstelle zwischen dem Benutzer und dem SmartHome bietet EVN eine Applikation für mobile Endgeräte an. In Abbildung 3 sind drei beispielhafte verschiedene Ansichten



Abbildung 3: Verschiedene Ansichten der Benutzeroberfläche der EVN SmartHome Applikation für Android

der Android Applikation zu sehen. In dieser werden nur Informationen, die unbedingt für die Aufgabenbewältigung nötig sind, angezeigt. So wirkt sie aufgeräumt und sollte für Benutzer, die den Umgang mit Smartphones gewöhnt sind, einfach zu bedienen sein.

2.4.3 Nest Thermostat

Im Gegensatz zu den bisher vorgestellten Lösungen stellt das nest Thermostat [Nes14] eine Benutzerschnittstelle mit gänzlich anderem Fokus dar. Das einzige Anwendungsgebiet des Thermostats ist die Steuerung der Heizung. Anfangs wird die Raumtemperatur, wie auch bei einem herkömmlichen Thermostat üblich, durch Auf- und Zudrehen geregelt. Weiterhin ist eine Steuerung über das Internet möglich, da das Gerät über das integrierte Wifi mit dem Heimrouter verbunden wird. Das nest Thermostat speichert die vom Endnutzer eingegeben Temperaturen für den jeweiligen Raum sowie die entsprechende Uhrzeit. Auf diese Art erlernt es die Gewohnheiten des Benutzers und kann, sobald es eine bestimmte Anzahl an Informationen gesammelt hat, automatisch die Raumtemperatur regulieren.



Abbildung 4: Übersicht der Benutzeroberfläche des nest Thermostats aus [Nes14]

Die direkte Interaktion mit dem Gerät ist äußerst intuitiv, da es die Form und das Aussehen eines herkömmlichen Thermostats besitzt. Der Benutzer muss die Bedienung des Geräts nicht erst erlernen, sondern kann es wie gewohnt verwenden und im späteren Verlauf der Nutzung von der Lernfunktion des Gerätes finanziell profitieren.

2.4.4 Forschungsprojekt MeRegioMobil

Im Rahmen des bereits im früheren Verlauf der Ausarbeitung erwähnten Förderungsprogrammes *E-Energy* des BMU wurde im Forschungsprojekt MeRegioMobil das Energy Management System (EMS) entwickelt. Die Hauptaufgaben des Systems, im Bezug auf den Energieverbrauch und das Energieangebot eines Smart Homes, sind das Regulieren

von Stromangebot und -Nachfrage, die Reduzierung des Energieverbrauchs sowie die Bereitstellung von System Services wie z.B. die Rückeinspeisung von Stroms des Smart Homes ins Netz bei drohenden Energieengpässen [BKS12]. Eine Verbindung zwischen dem EMS des Smart Homes und dem Bewohner stellt das sogenannte Energy Management Panel (EMP) dar. In Abbildung 5 ist die Benutzeroberfläche des EMS, das sogenannte EMP, zu sehen. Das Userinterface wird, zumindest im Rahmen des Energy Smart Home Lab des Karlsruher Institut für Technologie, auf einem Wandcomputer angezeigt.

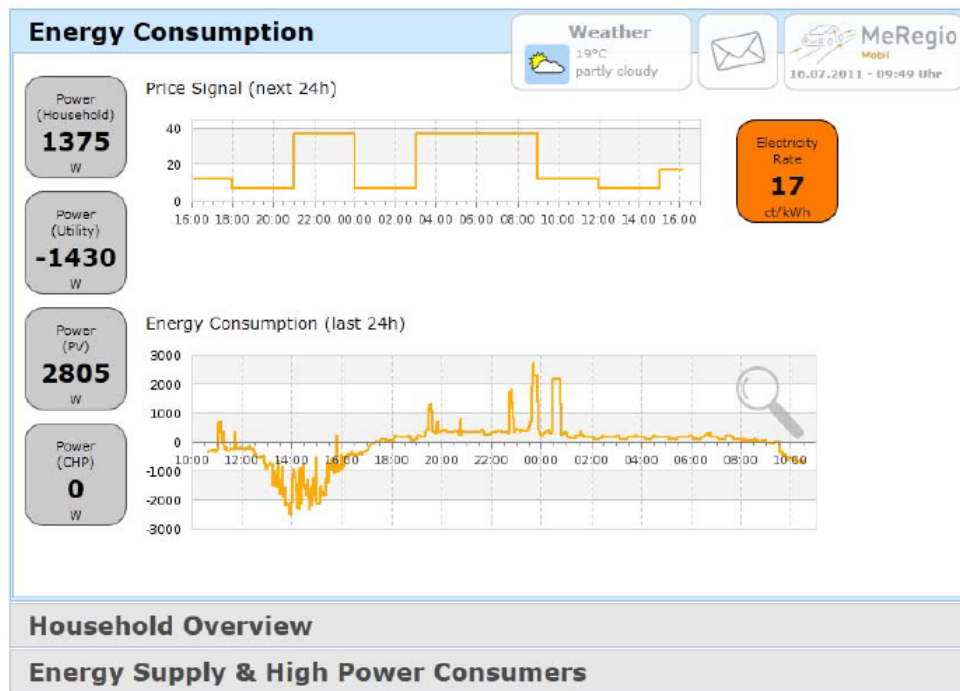


Abbildung 5: Übersicht der Benutzeroberfläche des EMP aus [BKS12]

Das EMP stellt dem Benutzer eine große Anzahl an Informationen über verschiedene Bereiche des Smart Homes bereit. Unter anderem hat der Benutzer mit dem EMP die Möglichkeit sowohl den aktuellen Energieverbrauch der einzelnen intelligenten Verbraucher einzusehen und zu analysieren als auch die mit dem Smart Home verbundenen Geräte zu steuern [BKS12]. Der Anwendungsbereich ist somit im Gegensatz zu zum *Nest Thermostat* und der Fernbedienung des *RWE Smart Homes* deutlich umfassender. Es stellt eine Gesamtlösung für ein Smart Home dar. Die in Abbildung 5 gezeigte Benutzeroberfläche zeigt bloß einen der drei möglichen Übersichtsbildschirme und dennoch ist bereits eine große Fülle an Informationen zu sehen. Diese sind von technischer Art und könnten daher für technikaverse Personen schwer zu verstehen sein.

3 Design

Das Design des alternativen Interfaces ist neben der Hardwareauswahl und Implementierung ein Hauptteil dieser Arbeit.

In diesem Abschnitt werden zunächst die Grundbegriffe der Mensch-System-Interaktion (MSI) erläutert, darauf folgend Teile der Richtlinien für die Mensch-System-Interaktion aus der Normenreihe *DIN EN ISO 9421: Ergonomie der Mensch-System-Interaktion* vorgestellt und verschiedene User Design Guidelines (UDG) präsentiert. Anschließend wird mit Hilfe der Richtlinien und der User Design Guidelines eine Designidee für einen ersten Prototypen entwickelt und präsentiert.

3.1 Richtlinien der Mensch-Computer-Interaktion

Die Richtlinien der Mensch-Computer-Interaktion (MCI) werden in der DIN EN ISO 9241 beschrieben. Daneben beinhaltet die Bezeichnung DIN EN ISO 9241 selbst weitere zahlreiche Informationen über die Norm. Sie ist auf Basis einer internationalen Norm der *International Standardisation Organisation*(ISO) entstanden, wurde anschließend in eine *European Norm*(EN) eingebunden und schließlich in die Deutsche Sprache übersetzt und durch das *Deutsche Institut für Normung e. V.* (DIN) übernommen. Somit stellt auch diese Norm einen freiwillig umzusetzenden Standard dar, um bei der Entwicklung neuer Soft- und Hardware dahingehend zu helfen, die Interaktion zwischen Mensch und Computer besser zu gestalten. Der Titel der DIN EN ISO 9241 wurde im Jahr 2006 von *Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten* in *Ergonomie der Mensch-System-Interaktion* geändert, um den bis dahin auf Software beschränkten Anwendungsbereich um den der Hardware zu erweitern [Hei12]. Infolgedessen können, zumindest Teile der Norm, für das vorliegende Problem genutzt werden.

3.1.1 Aufbau

Ursprünglich bildeten 17 Normteile die Gesamtnorm DIN EN ISO 9241. Im Laufe der Jahre wurden jedoch zahlreiche weitere Teile hinzugefügt. Für das Design eines Interfaces sind jedoch vor allem die Teilnormen *DIN EN ISO 9421-110: Grundlagen der Dialoggestaltung* und *DIN EN ISO 9421-11: Anforderungen an die Gebrauchstauglichkeit* von Bedeutung. Daher werden diese beiden Normen im weiteren Verlauf des Kapitels näher betrachtet.

3.1.2 Grundsätze der Dialoggestaltung

Im Teil DIN EN ISO 9421-110 werden die Grundsätze der Dialoggestaltung zwischen Mensch und interaktiven Systemen (Soft- & Hardware) beschrieben. Durch die Beach-

tung 7 Kriterien soll der Dialog zwischen Mensch und System verbessert werden. Diese sind Aufgabenangemessenheit, Selbstbeschreibungsfähigkeit, Erwartungskonformität, Lernförderlichkeit, Steuerbarkeit, Fehlertoleranz und Individualisierbarkeit. Folgende Beschreibungen orientieren sich an den Definitionen der Begriffe in der Norm [DIN06].

- **Aufgabenangemessenheit:** Ein interaktives System hat diese Eigenschaft, wenn es den Benutzer unterstützt, seine Aufgabe so zu erledigen, dass nicht die eingesetzte Technologie, sondern die Aufgabenerledigung im Mittelpunkt steht.
- **Selbstbeschreibungsfähigkeit:** Ein selbstbeschreibungsfähiger Dialog hat die Eigenschaft, dass für den Benutzer jederzeit offensichtlich ist, an welcher Stelle des Dialogs er sich gerade befindet und welche Handlungsoptionen er von dieser Stelle aus hat.
- **Erwartungskonformität:** Ein Dialog ist erwartungskonform, wenn die Bedienung und Informationsdarstellung eines Systems für den Benutzer den allgemeinen Konventionen bzw. seinen Erwartungen entspricht und dabei konsistent ist.
- **Lernförderlichkeit:** Ein Dialog wird als lernförderlich bezeichnet, wenn er den Benutzer beim Erlernen des interaktiven Systems Hilfestellungen gibt und ihm dadurch die Benutzung erleichtert. Dies kann zum Beispiel durch Verwendung von bekannten Formen und Mustern erreicht werden.
- **Steuerbarkeit:** Ein steuerbarer Dialog ermöglicht es dem Benutzer, in jedem Zustand des interaktiven Systems, den Dialogablauf zu beeinflussen. D.h. es sollten Start, Unterbrechungen, Wiederaufnahme und ein Widerrufen des letzten Dialogschritts möglich sein.
- **Fehlertoleranz:** Das Interaktive System soll selbst bei fehlerhaften Eingaben, ohne oder mit möglichst wenigen Korrekturen, in der Lage sein das gewünschte Ziel zu erreichen.
- **Individualisierbarkeit:** Ein Dialog ist individualisierbar, wenn der Benutzer die Möglichkeit hat, den Dialog an seine Bedürfnisse anzupassen.

3.1.3 Anforderungen an die Gebrauchstauglichkeit

In der Teilnorm DIN EN ISO 9421-11 werden drei Maße zur Bestimmung der Gebrauchstauglichkeit definiert. Diese sind Effektivität, Effizienz und Zufriedenstellung. Folgende Definitionen wurden aus der DIN EN ISO 9421-11 [DIN98] übernommen.

- **Effektivität:** Effektivität bezeichnet die Genauigkeit und Vollständigkeit, mit der Benutzer ein bestimmtes Ziel erreichen.

- **Effizienz:** Effizienz bezeichnet den im Verhältnis zur Genauigkeit und Vollständigkeit eingesetzten Aufwand, mit dem Benutzer ein bestimmtes Ziel erreichen.
- **Zufriedenstellung:** Zufriedenstellung bezeichnet die Freiheit von Beeinträchtigungen und positive Einstellungen gegenüber der Nutzung des Produkts.

3.1.4 Direct Manipulation / 9241-16

Direct Manipulation bezeichnet im Kontext der Mensch-Maschine-Interaktion einen Interaktionsstil, dessen Prinzipien erstmals 1983 von Shneiderman [Shn82] in der Fachliteratur erwähnt wurden. Catherine Plaisant und Shneiderman definieren *Direct Manipulation Interfaces* als „visual representation (metaphor) of the world of action with rapid, incremental and reversible actions; replacement of typing with pointing and selecting; and immediate visibility of results of actions.“ [Shn10] Nach dieser Definition handelt es sich um den Interaktionsstil der *Direkten Manipulation*, wenn eine sichtbare und kontinuierliche Repräsentation der Aufgabenobjekten und ihrer Aktionen vorhanden ist. Zudem sind Effekte der ausgeführten Aktionen sofort sichtbar und sollten reversibel sein.

Auch in DIN EN ISO 9216 findet sich der Begriff der direkten Manipulation wider. Im 16. Teil der Norm wird die *Dialogführung mittels direkter Manipulation* beschrieben. In dieser Teilnorm wird direkte Manipulation als „eine Dialogtechnik, durch die der Benutzer den Eindruck erhält, die Objekte am Bildschirm direkt zu bearbeiten, z. B. indem er mit Hilfe eines Zeigeelements auf sie zeigt, sie verschiebt und/oder ihre physikalischen Eigenschaften (oder Werte) verändert“ beschrieben.

3.2 Userinterface Design Guidelines

Die Richtlinien aus den Normen der letzten Abschnitte stellen standardisierte Empfehlungen aus Einsatzgebiet der MSI dar. Zudem haben Personen, die im Bereich des Designs von interaktiven Computer Systemen aktiv sind, versucht „gutes Design“ [Joh10] voranzubringen. So wurden im Laufe der Jahre Richtlinien für das Design neuer Produkte, Software sowie Hardware, veröffentlicht. Im weiteren Verlauf dieses Abschnitts wird eine Auswahl dieser, im Folgenden als *User Design Guidelines* (UDG) bezeichneten Empfehlungen, vorgestellt.

Aus dem Artikel *Design Rules Based on Analyses of Human Error* [Nor83] von Norman, einem Wissenschaftler aus dem Bereich des *Usability Engineering*, können folgende Lehren gezogen werden [Joh10]:

Es soll eine Form von Feedback geben, d. h. der Zustand des Systems sollte für den Benutzer jederzeit klar ersichtlich sein an welcher Stelle des Arbeitsschritts er sich gera-

de befindet, so dass Fehler vermieden werden können. Verschiedene Aktionen sollen mit möglichst unähnlichen Befehlsreihenfolgen umgesetzt werden, damit Ausführungsfehler vermieden werden können. Dem Benutzer soll die Möglichkeit gegeben werden Aktionen, falls noch möglich, rückgängig zu machen um ungewollte Fehler zu korrigieren. Zudem sollte das gesamte System eine konsistente Struktur als auch Befehlsart besitzen um die Zahl der Fehler bei der Ausführung zu minimieren.

Ähnliche Designempfehlungen über Feedback, Konsistenz des interaktiven Systems und der Möglichkeit Fehler rückgängig zu machen, finden sich in der Fachliteratur ebenfalls bei den Autoren Nielsen [NM90] sowie Shneiderman und Plaisant [Joh10].

3.3 Designidee

In den vorigen Abschnitten wurden die Normen DIN EN ISO 9241-110, DIN EN ISO 9241-11 und DIN EN ISO 9241-16 sowie bekannte *User Design Guidelines* (UDG) vorgestellt. Zudem wurde der Interaktionsstil der direkten Manipulation definiert und erläutert. Ein Teil dieser kann dabei helfen die Anforderungen, die an das alternative Interface gestellt werden, näher zu bestimmen und darauf folgend eine erste Designidee zu entwickeln. In diesem Abschnitt sollen daher folgende Fragen geklärt werden: Welche Funktionen sollen vorhanden und wie werden umgesetzt? Welche Form soll das Interface haben und warum? Warum wurde sich für bestimmte Art von Gestaltung entschieden ?

Das hier entworfene Gerät soll dem Benutzer in einfacher Art und Weise bei der Wahl des Beendigungszeitpunktes unterstützen. Dabei soll die dahinter eingesetzte Technologie für den Benutzer nicht sichtbar und Dialogmöglichkeiten möglichst simpel gestaltet sein, so dass die Erledigung der Aufgabe in den Mittelpunkt gestellt wird. Weiterhin soll die Form des alternativen User Interfaces selbst und auch die der Komponente des LED-Rings bewusst gewählt werden. Die Form wird daher der einer herkömmlichen Fernbedienung nachgeahmt und die LEDs des Rings sollen durch zusätzliche Beschriftung an die Form einer Uhr erinnern. So kann der Nutzer allein durch ihm bereits bekannte Dinge ein Verständnis über die Funktionsweise erhalten. Somit kann die Art der Dialoggestaltung von der Benutzerschnittstelle als *aufgabenangemessen* und *lernförderlich* im Sinne der DIN EN ISO 9241-110 [DIN06] aus Abschnitt 3.1.2 betrachtet werden.

Die Gesamtheit der Daten bzw. Rückmeldungen, die der Benutzer bei der Verwendung erhalten soll, werden über ein Display sowie über den Ring aus LEDs angezeigt. Damit das Feedback zuverlässig an den Benutzer übermittelt werden kann, ist es wichtig bei den auf dem Display und LED-Ring angezeigten Farben bzw. Farbkombinationen folgendes zu beachten. So sollte bei der Farbwahl Wert darauf gelegt werden, möglichst Farben zu wählen, die für den Benutzer leicht zu unterscheiden sind. Dazu sollten sehr distinktive Farben, wie in Abbildung 6 gezeigt, eingesetzt werden. Des Weiteren sollten Farbpaare, welche Farbenblinde nicht auseinanderhalten können, vermieden werden sowie Gegenfar-



Abbildung 6: Die sich am meisten unterscheidenden Farben aus [Joh10]

ben räumlich voneinander getrennt sein [Joh10]. Aus diesem Grund werden bei der Farbgebung der LEDs die Farben Rot, Grün, Gelb und Orange und bei dem Display Schwarz und Weiß verwendet. Die Farbe Orange bildet dabei eine Ausnahme von der Empfehlung von Johnson und wird nur eingesetzt um eine feinere Unterteilung zu ermöglichen.

Das alternative Interface wird auch für die Zielgruppe von älteren Personen sowie technisch weniger versierten Benutzern konzipiert. Daher wird angestrebt, besonders bezogen auf die Eingabemöglichkeiten, ein möglichst simples Interface zu entwickeln und damit eine intuitive Bedienung zu ermöglichen. Möglichst wenige Eingabemöglichkeiten könnten dazu beitragen, die gewünschte Funktionalität so simpel, wie es die zu bewältigende Aufgabe zulässt, bereitzustellen.

Unter der Berücksichtigung der vorangegangenen Überlegungen entstand der in Abbildung 7 gezeigte erste Prototyp aus Pappe.



Abbildung 7: Erster Prototyp aus Pappe. Auf dem oberen Teil des Prototyps ist das Display zu sehen. Auf diesem wird das ausgewählte Gerät, die Uhrzeit sowie die Kosten für den geplanten Beendigungszeitpunkt angezeigt. Weiter unten zu sehen ist ein LED Ring mit 24 LEDs, welche farbcodiert das Gütemaß des Stroms für die nächsten 12 Stunden anzeigen. Im Inneren des LED Rings befindet sich die eigentliche Steuerungseinheit, ein Eingabegerät das gleichzeitig Drehknopf und Taster ist.

4 Entwicklung

In diesem Abschnitt soll geklärt werden wie der Entwicklungsprozess des Interfaces ausgesehen hat und wie dieser umgesetzt wurde. Dieses Kapitel beinhaltet zunächst die Vorstellung des Soft- und Hardwareframeworks von Arduino. Darauf folgend werden die Komponenten, die bei der Umsetzung des alternativen Interfaces benutzt wurden, aufgelistet und wenn angemessen weitere Informationen gegeben sowie deren Wahl begründet. Anschließend wird, um die softwareseitige Umsetzung der Arbeit zu veranschaulichen, die generelle Programmstruktur von sogenannten Arduino-Sketches beschrieben und danach der auf dem Prototyp eingesetzte Code erklärt.

4.1 Arduino-Framework

Für dieses Projekt wurden sowohl für die physikalische Umsetzung als auch für die Programmierung der Hardware die Arduino-Plattform [Ard14] verwendet. Im Folgenden wird diese nun vorgestellt und die Gründe für deren Wahl dargelegt.

Die Softwareseite der Arduino-Plattform besteht aus zwei Teilen. Den einen Teil bildet die in Java geschriebene Entwicklungsumgebung *Arduino IDE*, welche unter der *GNU General Public License* lizenziert ist und somit frei verfügbar und quelloffen ist. Der andere Teil des Softwareangebotes sind die Bibliotheken zur Steuerung der Mikrocontroller welche ebenfalls unter der *GNU Lesser General Public License* veröffentlicht werden. Die Hardware besteht aus einem einfach aufgebauten Board auf dem sich ein Mikrocontroller ebenso wie digitale und analoge Ein- und Ausgänge befinden. Das Herzstück fast aller Arduino Boards stellt ein Atmel AVR-Kontroller aus der megaAVR-Serie, ein 8-Bit Mikrocontroller des Herstellers Atmel dar. Die Spannungsversorgung wird entweder über den Universal Serial Bus (USB) oder über ein externes Netzteil gewährleistet. Die Kommunikation wird über die serielle Schnittstelle realisiert, wobei die Programmierung über USB mit Hilfe eines *USB nach seriell Konverter* geschehen kann. Sämtliche Informationen über die Boards werden offen und frei verfügbar bereitgestellt.

Zum tieferen Verständnis der Funktionsweise und somit der Möglichkeit Änderungen vorzunehmen, werden dem Endnutzer die originalen Schaltpläne des Boardes in der Form von *Eagle CAD* Schema Dateien angeboten. Demzufolge wird das gesamte Arduino-Framework als Open Source Elektronik Prototyping-Plattform bezeichnet.

4.2 Komponenten

Der Markt für Komponenten von Arduino-Boards sowie Drittkomponenten wie verschiedenste Shields und Sensoren ist sehr unüberschaubar. Als Shields werden Erweiterungsplatinen für Arduino-Boards bezeichnet, die sich einfach auf das eigentliche Board stecken

lassen, wobei die Erweiterungen Sensoren oder Module sein können. Betrachtet man allein die Anzahl der am Markt verfügbaren Arduino-Boards, wird die Unübersichtlichkeit des Marktes schnell klar. Neben 20 offiziellen Referenz-Boards, werden hunderte zur Arduino-Plattform kompatible Varianten, die sich in Funktionalität, Preis sowie Einsteigerfreundlichkeit unterscheiden, angeboten. Auch für jede Art von Shield sowie für jede Art von Sensoren existieren unzählige Alternativen.

Durch die in Abschnitt 1.1 definierten Ziele ergeben sich Ansprüche an das Interface und damit auch an die zu verwendende Hardware. Auch die in Kapitel 3 behandelten *Userinterface Design Guidelines* [Joh10] können für die Wahl der Komponenten herangezogen werden. Zudem ist der Zeitrahmen, in der dieses Projekt umgesetzt wurde, ein wichtiger Faktor. Es muss deshalb eine Auswahl darüber stattfinden, welche Komponenten verwendet werden sollen.

Im folgenden Abschnitt wird daher die Auswahl der Hauptkomponenten, welche in *Kommunikation, Ein- und Ausgabe* sowie *Hardwarebasis* untergliedert ist, begründet. Im Anschluss wird, auf Basis der Verbrauchseigenschaften der einzelnen Komponenten, eine Schätzung der zu erwartenden Laufzeit des Prototyps aufgestellt.

Hardwarebasis:

Das Arduino-Board stellt das Herzstück des alternativen Interfaces dar. Es muss den richtigen Typ sowie eine ausreichende Anzahl an Inputs bzw. Outputs bereitstellen und dabei noch in einem akzeptablen Preisrahmen bleiben.

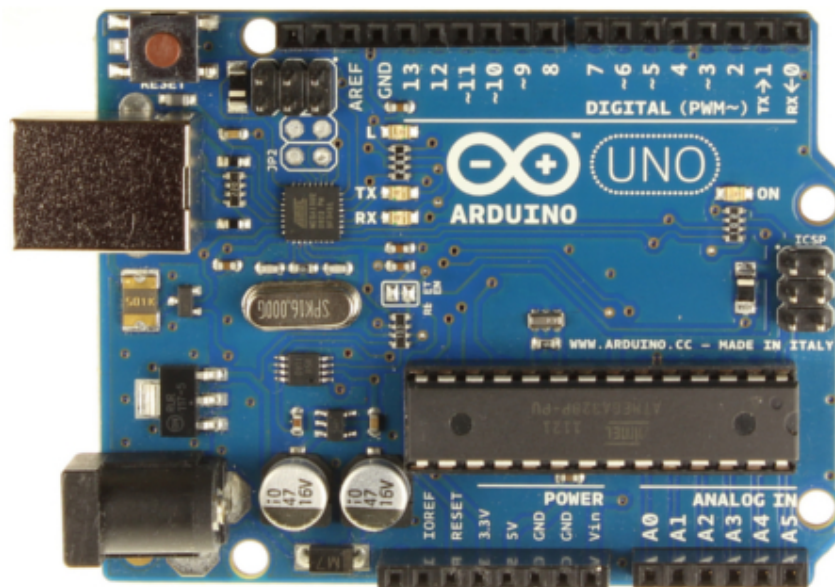


Abbildung 8: Obere Seite des Arduino Uno R3

Die Auswahl des Boards fiel auf das in Abbildung 8 präsentierte *Arduino Uno R3*. Der Hauptgrund dafür ist, dass das Arduino R3 das mit Abstand meistverkaufte Board ist und somit zwei Vorteile mit sich bringt. Durch seine Beliebtheit existiert eine immense Anzahl an Bibliotheken, die sowohl von Benutzern als auch Entwicklern der Drittkomponenten

geschriebenen wurden und ohne weitere Modifikationen am Quelltext mit dem Arduino Uno R3 kompatibel sind. Des Weiteren ist das meiste verfügbare Zubehör interoperabel zum R3 und kann ohne großen Aufwand mit diesem verbunden werden

Ein- und Ausgabe:

Die hier vorgestellten und in Abbildung 9 gezeigten Komponenten sind für die jegliche In- und Outputs des Prototyps verantwortlich.

Die Möglichkeiten der Eingaben sollen, wie in 3.3 bereits beschrieben, möglichst einfach gehalten werden ohne dabei die gewünschte Funktionalität einzuschränken. Der *Rotary Encoder* bzw. Drehgeber Grove eignet sich sehr gut, da er einen integrierten Pushbutton besitzt. So können die Eingaben entweder durch eine Drehen des Drehgebers oder durch ein Drücken auf eben diesen erfolgen. Zudem hat er wie alle Komponenten von Grove den Vorteil, dass mit einer Bibliothek geliefert wird und nicht verlötet werden muss.

Die Ausgabe teilt sich auf zwei Komponenten auf, dies ist zum einen ein monochromes OLED-Display und zum anderen ein LED-Ring mit 24 RGB-LEDs. Das Display von Grove wurde anderen Alternativen vorgezogen, da es im Gegensatz zu LCD-Displays einen höheren Kontrast bietet sowie weniger Energie verbraucht. Zudem hat die Zugehörigkeit zur Grove-Serie den Vorteil, dass es ohne Lötarbeiten angeschlossen werden kann. Für die Wahl des *Neopixel* LED-Ring von *Adafruit* spricht auf der einen Seite die Anzahl der LEDs und ihre Anordnung in Form einer Uhr und auf der anderen Seite die gute Konfigurierbarkeit durch die umfangreiche mitgelieferte Bibliothek. Durch die Nutzung von zwei verschiedenen Ausgabegeräten ist es außerdem möglich dem Benutzer ein redundantes Feedback zu geben und so die *Selbstbeschreibungsfähigkeit*, die im Rahmen von Abschnitt 3.1.2 bereits behandelt wurde, zu erhöhen.

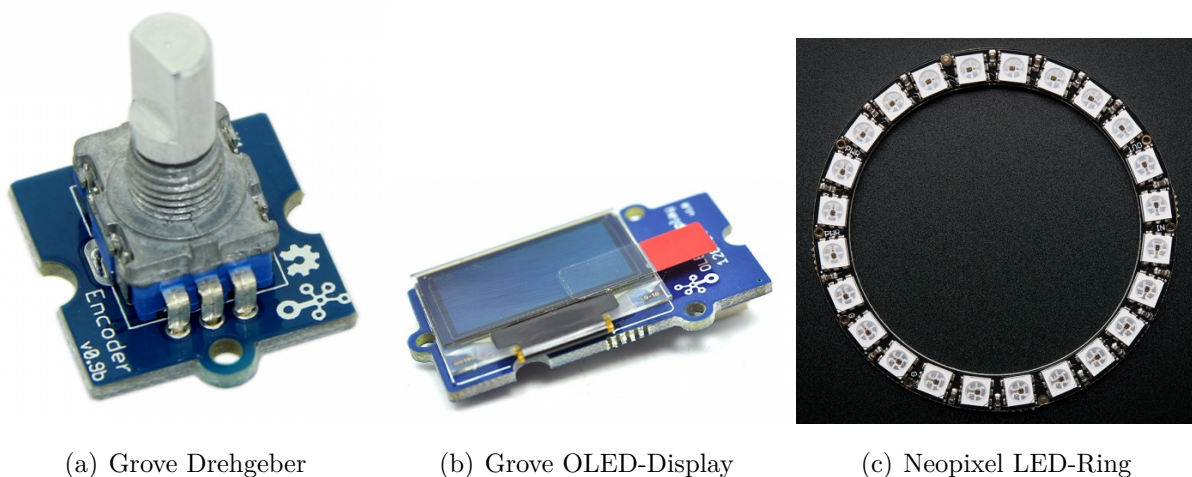


Abbildung 9: Im Prototyp verwendete Ein- und Ausgabekomponenten

Kommunikation:

Es gibt eine Vielzahl von Alternativen, die für die drahtlose Datenübertragung bei Projek-

Tabelle 1: Vergleich drahtloser Module bzw. Techniken, Daten aus den entsprechenden Datenblättern ([Zig14], [Dig14], [Cis14], [BLE14], [Sem14], [RN14])

	ZigBee	Z-Wave	XRF	BLE 4.0	nRF24	Wifi
max. Verbrauch Rx	23 mA	24 mA	23,8 mA	24.6 mA	12 mA	40 mA
max. Verbrauch Tx	36 mA	36 mA	36,2 mA	31.6 mA	12 mA	240 mA
Datenrate (mbit/s)	0,25	0,04	0,25	1	2	24
Verschlüsselungsalgorithmus	128-bit AES	keine	128-bit AES	128-bit AES	keine	WPA-2 PSK u.A.
Frequenzbereich	2,4 GHz	868 MHz	868 MHz	2,4 GHz	2,4 GHz	2,4 GHz
Reichweite Indoor	max. 15m	ca. 30m	ca. 30m	ca. 10m	ca. 15m	>30m
Stückpreis (€)	ca. 22	ca. 12	14,13	ca. 20,80	<5	28,95

ten für Arduino, eingesetzt werden können. In Tabelle 1 findet sich eine Auswahl verschiedener Module, die für das vorliegende Problem in Betracht gezogen werden können. Je nach dem in welchem Aufgabenbereich ein drahtloses Modul eingesetzt werden soll, bestehen unterschiedliche Ansprüche und so wird auf jeweils andere technische Eigenschaften Wert gelegt. Es gibt weit mehr Aspekte unter denen die verschiedenen Module verglichen werden könnten, jedoch sind für das im Rahmen dieser Arbeit umgesetzte Projekt vor allem die Aspekte der Tabelle 1 wichtig.

Der erste Vergleichspunkt, *max. Verbrauch Rx*, bezeichnet die maximale Leistungsaufnahme des Moduls während des Empfangs von Daten. Entsprechend wird unter *max. Verbrauch Tx* die maximale Leistungsaufnahme des Moduls während des Sendens von Daten verstanden. Diese beiden Punkte sind besonders wichtig, da es sich bei dem umgesetzten Interface um ein mobiles Endgerät handelt und somit der Stromverbrauch während der Datenübertragung möglichst gering gehalten werden soll.

Es wird davon ausgegangen, dass das alternative Interface maximal 10 Minuten pro Tag aktiv ist. Nur in einem geringen Teil dieses schon sehr beschränkten Zeitraums senden oder empfangen die Funkmodule. Auf Grund dieser Annahmen kann die Aussage getroffen werden, dass nach diesen beiden Vergleichsmerkmalen betrachtet, mit Ausnahme des Wifi-Moduls, alle Module sich in einem Bereich, der für die Umsetzung akzeptabel wäre, befinden. Genauere Berechnung zu der Akkulaufzeit des gesamten Systems werden noch

an späterer Stelle in diesem Kapitel präsentiert.

Der dritte Vergleichsaspekt, die *Datenrate* steht für die maximal mögliche Übertragungsrate, die zwischen Sender und Empfänger möglich sind.

Wie später noch in 4.3 näher ausgeführt empfängt das Interface nach Einschalten einen Array, in dem sich neben der Uhrzeit das Gütemaß des verfügbaren Stroms in Form von 24 Zahlen im Wertebereich von 1 bis 100 befindet. Werden diese Daten in Form eines einfachen Austauschformat wie JSON übertragen, ist dieses Array lediglich 117 Byte groß. Auf Grund dieser äußerst niedrigen Ansprüche ist jedes der Module bezüglich der Datenrate geeignet.

Als Viertes wird unter dem Punkt *Verschlüsselungsalgorithmus* betrachtet, ob und gegebenenfalls wie, bei der jeweiligen Funktechnik eine Verschlüsselung bei der Übertragung der Daten möglich ist. Dies ist von grundlegender Bedeutung da zum einen Daten, die zwischen den verschiedenen Akteuren übertragen werden, datenschutztechnisch betrachtet hochsensibel sind. Zum anderen besteht in eng besiedelten Gebieten, falls keine weiteren Sicherheitsmaßnahmen wie Verschlüsselung getroffen werden, die Gefahr, dass Signale aus benachbarten Smart Homes von nicht gewollten Geräten empfangen oder manipuliert werden könnten.

Bis auf die Lösungen *Z-Wave* und *nRF24* bieten alle Module eine Verschlüsselung an. Diese wären somit unter dem Gesichtspunkt der Sicherheit geeignet.

Der *Frequenzbereich* ist im Rahmen dieser Arbeit aus zwei Gründen relevant. Entsprechend §55 Abs. 2 des Telekommunikationsgesetz (TKG), werden von der *Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen* (BNA) Allgemeinurteilungen, die den Nutzerkreis und die Art der Nutzung bestimmen, veröffentlicht. Durch starke Nutzung des ISM 2,4 Gigahertz-Bandes kann leicht zu Störungen zwischen den verschiedenen Geräten kommen. Aus diesem Grund war ein von Anfang gewünschtes Ziel, dass die drahtlose Kommunikation auf einem nicht überfüllten Frequenzband stattfinden soll. Ein Solches stellt das 868 Mhz Band dar.

In diesem Punkt spricht die Nutzung des Short-Range-Devices-Bands 868 MHz für Module *XRF* und *Z-Wave*. Die Nutzung dieses Bands ist jedoch durch die Allgemeinurteilungen Vfg 43 / 2012 [BNA14] der Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen (BNA) reguliert, so dass einige Punkte beachtet werden müssen. Beide eben genannten Module laufen auf 868,3 MHz und fallen somit in den Frequenzbereich von 868,0 MHz bis einschließlich 868,6 MHz. Für diesen Bereich gibt es zwei Einschränkungen: Zum einen die *Maximale Magnetische Feldstärke* in Höhe von 25mW, zum anderen darf der *Arbeitszyklus* des Moduls 1% nicht übersteigen falls keine weiteren *Frequenzzugangs- und Störungsminderungstechniken* eingesetzt werden. Nach den Datenblättern der *XRF* [Cis14] und *Z-Wave* [Dig14] Module beträgt die maximale magnetische Feldstärke 10mW bzw. 1mW. Der *Arbeitszyklus* oder auch als relative Frequenzbe-

legungsdauer bezeichnet, „ist definiert als anteilmäßiger aktiver Sendebetrieb innerhalb einer Zeitdauer von einer Stunde zu einem beliebigen Zeitpunkt“ [BNA14]. Das bedeutet konkret, dass die im Smart Home eingesetzte Module maximal 36 Sekunden pro Stunde Daten senden dürfen. Da wie vorherig angesprochen die Datenmenge sehr gering ist, kann diese Beschränkung problemlos umgesetzt werden.

In der sechsten Zeile der Tabelle wird die *Reichweite Indoor* miteinander verglichen. Sie bezeichnet den Abstand, den die Sendeempfänger maximal zueinander haben dürfen um eine fehlerfreie Kommunikation gewährleisten zu können.

Die Module die in Frage kommen sind diejenigen mit einer hohen Reichweite. Die höchste Reichweite bietet mit über 30 Metern das Wifi, gefolgt von den beiden auf 868 MHz funkenden Modulen *Z-Wave* und *XRF*. Aufgrund der Frequenz könnten die beiden zuletzt genannten Module jedoch unter realen Bedingungen eine höhere Reichweite besitzen.

Der letzte Vergleichspunkt ist der *Preis des Moduls* in €. Wie in Kapitel 1.1 im Rahmen der Zielsetzung angesprochen, soll der maximale Preis für das gesamte Interface den Betrag von 100€ nicht übersteigen. Die Preise haben einen Preisbereich von 5€ bis ca. 29€ und machen somit einen nicht zu verachtenden Teil des Gesamtpreises aus.

Bei Verfügbarkeit wurden die Preise über die Webseite des Exp-Tech Shops [Exp14] bestimmt, ansonsten über verschiedene Angebote auf Ebay eine Schätzung durchgeführt.

Auch bei der in Abschnitt 2.4.1 vorgestellten Fernbedienung des *RWE Smart Homes* wird auf eine Funktechnik im 868,3 MHz Bereich gesetzt und somit kann angenommen werden, dass die Technik ausführlich erprobt wurde. Des Weiteren sprechen sowohl technische als auch aus praktische Gründe für das Modul von Ciseco, das *XRF*. Der maximale Verbrauch zum Senden und Empfangen sowie die Datenrate befinden sich in einem höchst akzeptablen Bereich. Durch die Nutzung der Frequenz von 868,3 MHz ergibt sich, allein durch die niedrigere Frequenz, eine höhere Reichweite als bei den verglichenen 2,4 GHz Modulen. Die Sicherheit der Daten wird durch eine einfach aktivierbare Verschlüsselung gewährleistet und zudem ist die Verbindung zum Arduino-Board leicht möglich, da es über das weit verbreitete XBee Interface angeschlossen werden kann.

Abschätzung der Laufzeit im Akkubetrieb:

Die in Tabelle 2 gezeigten Werte sind aus den entsprechenden Datenblättern entnommen. Dabei wurden jeweils die maximalen Werte genommen und diese leicht aufgerundet um bei den Komponenten, dessen Standby-Verbrauch sich im Bereich von wenigen μA befindet, diesen außer acht lassen zu können.

Aus einem Verbrauch von 230 Milliampere (mA) im Betrieb und einem Standby-Verbrauch im Sleep-Modus von 35 mA lassen sich die Bereitschaftszeiten berechnen. Bei einer Betriebszeit von 15 Minuten pro Tag ergibt sich ein Verbrauch von $23,75 \cdot 35 \text{ mA} + 0,25 \cdot 230 \text{ mA} = 888,75 \text{ mA}$ und bei 30 Minuten pro Tag dementsprechend $23 \cdot 35 \text{ mA} + 0,5 \cdot 230 \text{ mA} =$

937,5mA. Für die mobile Stromversorgung wurde ein Lithium-Ionen-Akku mit einer Ladekapazität von 2000 Milliamperestunden verwendet, so dass bei den beiden oben genannten Szenarien jeweils eine Bereitschaftszeit von über 2 Tagen realistisch ist.

Tabelle 2: Vergleich des Stromverbrauchs der Komponenten im Betrieb und Standby

	Verbrauch im Betrieb	Verbrauch im Standby
Uno R3	50mA	35 mA
LED-Ring	100mA	-
OLED-Display	20mA	-
Drehgeber	20mA	-
XRF	40mA	-
Gesamt	230mA	35mA

4.3 Umsetzung

Jegliche Programme, im weiteren Verlauf wie für Arduino-Programme typisch ebenfalls als Sketche bezeichnet, wurden in der Arduino IDE in der Version 1.0.5 [Ard14] bzw. in Microsoft Visual Studio Professional 2013 mit Hilfe des *Visual Micro*² Addons geschrieben.

4.3.1 Allgemeine Programmstruktur

Der grundlegende Aufbau eines Arduino Programms ist relativ simpel und teilt sich in mindestens die Blöcke *setup()* und *loop()* auf. Zusätzlich können vor dem *setup()*-Block mit dem Befehl *#include* Bibliotheken eingebunden werden um die Funktionalität des Programms zu erweitern. Die im Programmcode 1 verwendete Bibliothek *SoftwareSerial* ermöglicht eine serielle Kommunikation über zwei beliebige digitale Pins des Arduinos, denn standardmäßig haben Arduino-Boards lediglich eine in die Hardware integrierte Unterstützung für serielle Kommunikation über die Pins 0 und 1.

```

1
2 #include <SoftwareSerial.h>
3 SoftwareSerial mySerial(10, 11);
4
5 void setup()
6 {
7     begin.mySerial(9600);
8     pinMode(9,OUTPUT);
9 }
10 void loop()
```

²nutzt ebenso den Compiler der Arduino IDE 1.05

```
11 {  
12   digitalWrite(9, HIGH);  
13   delay(500);  
14   digitalWrite(9, LOW);  
15 }
```

Programmcode 1: Elemente eines Sketches

Notwendig für das Ausführen eines Programms sind die *setup()* und *loop()* Funktionen. Fehlt eine der beiden Funktionen kann das Programm nicht kompiliert werden. Hierbei enthält *setup()* Anweisungen die, sobald der Arduino eingeschaltet wurde, einmalig ausgeführt werden. Eine erneute Ausführung findet nur statt, wenn die Stromzufuhr unterbrochen und anschließend aufs Neue hergestellt oder das Board zurückgesetzt wird.

Nach der *setup()* wird die *loop()* Funktion aufgerufen. Diese beinhaltet Programmcode, der in einer unendlichen Schleife solange ausgeführt wird, bis dem Arduino Board kein Strom mehr zur Verfügung steht.

Zur Vereinfachung und für eine höhere Wiederverwendbarkeit können optional eigene Funktionen definiert werden. Ein Beispiel für die Nutzung eines Programms mit einer eigens geschriebenen Funktion findet sich in Programmcode 2 ab Zeile 12. Die Funktionalität des folgenden Programmcodes entspricht exakt der des im vorigen Abschnitt gezeigten Codes.

```
1 #include <SoftwareSerial.h>  
2 SoftwareSerial mySerial(10, 11);  
3  
4 void setup() {  
5   begin.Serial(9600);  
6   pinMode(9, OUTPUT);  
7 }  
8 void loop()  
9 {  
10   blink();  
11 }  
12 static void blink()  
13 {  
14   digitalWrite(9, HIGH);  
15   delay(500);  
16   digitalWrite(9, LOW);  
17 }
```

Programmcode 2: Beispiel selbstdefinierter Funktion

4.3.2 Umsetzung

Im letzten Abschnitt wurde der allgemeine Aufbau eines Arduino-Sketches vorgestellt um im folgenden Teil die Funktionsweise des eigens entwickelten Sketches erläutern zu können. Daher soll zunächst der grobe Aufbau des Programms aufgezeigt und anschließend einzelne, für den Programmablauf wichtige, Funktionen erklärt werden.

Aufbau & Ausführungsverhalten: Das Programm ist wie folgt aufgebaut. Im Programmcode werden als Erstes die vom jeweiligen Hersteller bereitgestellten Bibliotheken geladen, damit der Drehgeber, das OLED-Display sowie der NeoPixel Led-Ring in der Arduino-IDE überhaupt angesprochen werden können. Das Funkmodul kommuniziert über eine serielle Schnittstelle mit dem Arduino und benötigt daher keine zusätzliche Bibliothek. Anschließend werden vier *static unsigned char*-Arrays definiert. Diese repräsentieren die später im Programm genutzten vier Symbole der Verbraucher Waschmaschine, Trockner, Spülmaschine und Auto. Anschließend folgt die Initialisierung verschiedener global sichtbarer Variablen und anknüpfend die *setup()*-Funktion. Dort werden hauptsächlich der Drehgeber, das Display sowie der LED-Ring initialisiert, weitere Details dazu finden sich in den Kommentaren des Quelltexts.

Der interessante Teil befindet sich in der *loop()*-Funktion, denn dort werden die eigens programmierten Funktionen aufgerufen.

```
1 void loop() {  
2     if (programState==0) drawBitmaps();  
3     if (programState==1) choseTime();  
4     if (programState==2) confirmTransmission();  
5 }
```

Programmcode 3: *loop()*-Funktion des finalen Sketches

Wie im oberen Code-Snippet zu sehen, werden bei Benutzung des Userinterfaces abhängig von dem Zustand des Programms, welcher durch die Variable *programState* repräsentiert wird, eine von drei Funktionen aufgerufen.

Programmmzustand 0: Zu Beginn des Programms wird die Funktion *drawBitmaps()* aufgerufen, da die Zustandsvariable anfangs mit einem Wert von Null initialisiert wird. Innerhalb dieser Funktion kann mit Hilfe des Drehgebers eines der Geräte ausgewählt werden. Dabei merkt sich das Programm die Position an der es sich innerhalb der vier Verbraucher befindet und übergibt diese der eigentlichen Zeichnungsfunktion. Hat der Nutzer beispielsweise die Waschmaschine ausgewählt, dreht nach links und anschließend nach rechts, wird abermals die Waschmaschine auf dem Display angezeigt. Sobald die Auswahl des Gerätes abgeschlossen ist und durch Druck des Pushbuttons bestätigt wurde, nimmt die Variable *programState* den Wert von eins und das Programm geht in den nächsten Zustand über.

Programmmzustand 1: An dieser Stelle im Programm kann der Benutzer die den Beendi-

gungszeitpunkt für das im Programmzustand 0 gewählten Geräts, bestimmen. Dies geschieht innerhalb der *choseTime()*-Funktion. Die aktuelle Auswahl für den Beendigungszeitpunkt auf zwei Arten angezeigt. Zum einen auf dem OLED-Display in Form einer digitalen Uhr, zum anderen blinkt die der aktuellen Selektion entsprechenden LED des LED-Rings in blau. Verändert der User den Zeitpunkt für die Beendigung ändert sich sowohl die auf dem Display angezeigte Zeit als auch die aktuell blinkende LED. So sollte für Benutzer die derzeitige Auswahl jederzeit klar sein.

Programmmzustand 2: Nachdem der Benutzer einen Beendigungszeitpunkt ausgewählt hat, wird der Variable *programState* ein Wert von zwei zugewiesen und die Funktion *confirm-Transmission* gestartet. In dieser wird die Auswahl per Funkmodul an den Empfänger übertragen. Ist das Senden erfolgreich, nehmen alle LEDs des Rings die Farbe grün an. Ein erneutes Drücken des Pushbuttons führt dazu, dass das Gerät wieder in den Anfangszustand zurückgesetzt wird und der Beendigungszeitpunkt für ein weiteres Gerät eingerichtet werden kann.

Einige der Funktionen, die das Programm nutzt, wurden nicht in Eigenarbeit programmiert, sondern greifen auf die Bibliotheken der jeweiligen Komponenten zurück. Bis auf den Hersteller des Funkmoduls bieten alle Fabrikanten für ihre Komponenten vorgefertigte Bibliotheken an, welche die Benutzung durch die Bereitstellung von Funktionen erleichtern. In dem folgenden Ausschnitt aus Quellcode wird die Zuordnung der Bibliotheken zu den entsprechenden Komponenten gezeigt.

```
1 #include <Encoder.h>           //Drehgeber
2 #include <TimerOne.h>         //Drehgeber
3 #include <Wire.h>             //Drehgeber & Display
4 #include <SeedOLED.h>         //Display
5 #include <avr/pgmspace.h>      //Display
6 #include <Adafruit_NeoPixel.h> //Led-Ring
```

Programmcod 4: Komponenten und ihre Bibliotheken

Der gesamte Quellcode befindet sich auf der zur Bachelorarbeit beiliegenden CD-ROM.

5 Evaluation

Zu Beginn der Ausarbeitung wurden in Abschnitt 1.1 sechs verschiedene Ziele hinsichtlich der Eigenschaften des entwickelten Smart Home Userinterfaces gesetzt. Diese dort definierten Ziele sollen nun auf ihre Erfüllung sowie den Grad der Erfüllung überprüft und bewertet werden.

Ziel 1 - Intuitive Bedienung für technikaverse Personen

Das Interface wurde in Form einer Fernbedienung entwickelt. Durch diese Form sollte dem Benutzer schon bei Sichtung des Gerätes klar sein, dass es zur Fernsteuerung bestimmter Geräte verwendet werden kann. Weiterhin gibt es lediglich die Eingabemöglichkeit über einen Drehknopf mit Pushbutton-Funktion. Der Knopf kann zur Eingabe nach links oder rechts gedreht sowie gedrückt werden. Durch diese simplen Interaktionsmöglichkeiten wird die intuitive Bedienung gefördert. Wie in Abschnitt 3.3 beschrieben, wurde für die Benutzerschnittstelle eine Fernbedienung und für den LED-Ring eine Uhr als Formvorlage gewählt. Dies hilft könnte dem potentiellen Benutzern im Sinne der *Lernförderlichkeit* die Funktionsweise des Gerätes leichter zu erlernen. Diese Aussagen über die Einfachheit der Bedienung sind jedoch unter dem Vorbehalt zu betrachten, dass das Interface nicht mit Hilfe von Probanden oder anderen geeigneten Maßnahmen getestet wurde.

Ziel 2 - Feedback

Das Design und die Umsetzung geschahen unter Berücksichtigung der Grundsätze von Shneiderman's *Direct Manipulation Interfaces* [Shn10]. Schaltet der Benutzer das Gerät ein, erhält er augenblicklich zwei verschiedene Rückmeldungen. Zum einen erleuchtet, sobald das Gerät die Daten vom Raspberry Pi empfangen hat, der LED-Ring, zum anderen sieht der Nutzer auf dem Display das derzeit ausgewählte Gerät. Hat der Nutzer das Gerät durch einen Druck auf den Pushbutton ausgewählt, blinkt auf dem LED-Ring die LED, die der derzeitigen Zeitauswahl entspricht, so dass er sich bewusst ist welche Uhrzeit er momentan gewählt hat. Nach der Auswahl des Beendigungszeitpunktes bekommt der Nutzer eine Rückmeldung darüber ob seine Selektion erfolgreich übertragen wurde, indem der gesamte LED-Ring grün leuchtet. Somit erhält der Benutzer während der gesamten Zeit, in der eine Interaktion mit der System stattfindet, Rückmeldungen.

Ziel 3 - Datenschutz/Datensicherheit

Sämtliche Daten, die zwischen dem Arduino und dem Raspberry Pi mittels des XRF Moduls übertragen werden, werden hardwareseitig auf beiden Seiten vom jeweiligen Modul mit dem *Advanced Encryption Standard mit 128 Bit Schlüssellänge* (AES-128) verschlüsselt. Dadurch sollte eine Manipulation der Daten durch Dritte deutlich erschwert werden.

Ziel 4 - Vermeidung von Interferenzen & Erhöhung der Reichweite

Als Lösung für die kabellose Übertragung wurde ein Bauteil des Herstellers Ciseco mit

dem Namen XRF [Cis14] gewählt. Durch die Nutzung einer Frequenz von 868,3 MHz, entstehen sowohl Vor- als auch Nachteile. Die im Gegensatz zu den anderen Lösungen niedrigere Frequenz wirkt sich zwar negativ auf die Bandbreite aus, stellt jedoch bei der hier behandelten Problemstellung kein Hindernis dar, da die zu übertragenden Daten in einem Bereich von unter einem Kilobyte liegen. Positive Aspekte der gewählten Frequenz sind die damit einhergehende Erhöhung der Reichweite sowie die geringe Auslastung des Frequenzbandes.

Ziel 5 - Anschaffungskosten

Durch die Nutzung von Arduino Hardware war es möglich die Kosten für das gesamte Projekt verhältnismäßig niedrig zu halten. Sieht man vom Bau des Gehäuses für das alternative Userinterface ab, so ergibt sich ein Endpreis von unter 100€.

Ziel 6 - Funktionen des Userinterfaces Theoretisch ist die entwickelte Mensch-Maschinen-Schnittstelle in der Lage die in der Zielstellung gesetzten Funktionalitäten bereitzustellen. Es wurde jedoch kein Test mit Datenübertragung von einem anderen Endgerät wie dem des Raspberry Pi's durchgeführt.

6 Zusammenfassung & Ausblick

In diesem Kapitel soll die Arbeit rückblickend betrachtet und ein Ausblick zu weiteren Fragestellungen und zukünftigen Arbeiten gegeben werden.

6.1 Fazit

Im Rahmen dieser Arbeit wurde aufgezeigt wie und mit welchen Mitteln ein alternatives Userinterface für Smart Homes entworfen und implementiert werden kann. Dazu wurde zuerst die Notwendigkeit für ein solches Vorhaben begründet, folgend die Grundlagen der Mensch-Computer-Interaktion sowie Normen aus dem Bereich der Mensch-System-Interaktion vorgestellt. Daraufhin wurden verschiedene sich am Markt befindliche Beispiele von Benutzeroberflächen von Smart Homes präsentiert. Anschließend wurde unter der Beachtung von der durch Normen gegebenen Empfehlungen sowie durch die Berücksichtigung von Userinterface-Design-Guidelines, ein alternatives Userinterface auf Basis der Arduino-Plattform entwickelt. In Der Theorie erfüllt dieses die in der Einleitung formulierten Zielansprüche. Um eine belastbare Aussage treffen zu können, fehlt es jedoch an Tests im realen Umfeld. Es existieren, wie in Kapitel 5 gezeigt, bei jedem der Punkte noch weiteres Möglichkeiten zur Verbesserung. Diese werden nun vorgestellt.

6.2 Ausblick

Im Kontext dieser Bachelorarbeit wurde ein alternatives Userinterface für Smart Homes design und implementiert. Zu diesem Thema existieren jedoch nach wie vor eine Reihe von ungeklärten oder unzureichend geklärten Fragestellungen, so dass es im Hinblick auf weitere Arbeiten viele Möglichkeiten gibt.

Eine davon wäre die Beobachtung und Analyse der Nutzung des Gerätes unter realen Bedingungen. Mit Hilfe von verschiedenen Personengruppen könnte in Tests ermittelt werden, wie einfach oder schwer die Benutzung des Interfaces tatsächlich ist. Mit den daraus gewonnenen Daten könnte anschließend ein verbesserter Prototyp entwickelt werden.

Weiterhin wurden auf Grund des begrenzten zeitlichen Rahmen bei der Auswahl der Komponenten fast ausschließlich Produkte aus der Grove Reihe gewählt. Diese müssen lediglich über einen Adapter mittels einer Steckverbindung mit dem Arduino-Board verbunden werden und benötigen kein Verlöten, so dass nach Erhalt der Komponenten relativ zeitnah mit der Programmierung begonnen werden konnte. Ein Nachteil von Grove-Komponenten ist jedoch, dass diese größer und meist teurer als in der Funktionalität identische Alternativen sind. So besteht in diesem Bereich, hinsichtlich des Gesamtpreises und der Größe noch Optimierungspotential.

Wie in Kapitel 4 erläutert ist der Standby-Verbrauch sehr hoch und könnte durch einen Austausch des Arduino-Boards auf einen Bruchteil reduziert werden.

Zukünftige Arbeiten in dem Themenkomplex dieser Bachelor-Thesis werden weitere Fragestellungen zu Tage bringen. Die vorliegenden Punkte sollen daher keinen Anspruch an Vollständigkeit erheben.

Literatur

- [AGE13] *Struktur der Stromerzeugung in Deutschland 2013*. http://www.ag-energiebilanzen.de/index.php?article_id=29&fileName=ageb_infografik_02_2013_stromerzeugung_2013.pdf, 2013. [Online; Zugang: 30.01.2014].
- [Ard14] *Arduino Homepage*. <http://arduino.cc/>, 2014. [Online; Zugang: 30.01.2014].
- [BKS12] BECKER, B., A. KELLERER und H. SCHMECK: *User interaction interface for Energy Management in Smart Homes*. In: *Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES*, Seiten 1–8, Jan 2012.
- [BLE14] *Datenblatt Bluetooth Low Energy SoC CC2540F128, CC2540F256*. <http://www.ti.com.cn/cn/lit/ds/symlink/cc2540.pdf>, 2014. [Online; Zugang: 31.03.2014].
- [BMW13] BMWI: *Energie in Deutschland - Trends und Hintergründe zur Energieversorgung*. <http://www.bmwi.de/Dateien/Energieportal/PDF/energie-in-deutschland>, 2013. [Online; Zugang: 10.02.2014].
- [BNA14] *Allgemeinzuteilung Vfg 42/2012*. http://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Sachgebiete/Telekommunikation/Unternehmen_Institutionen/Frequenzen/Allgemeinzuteilungen/2012_43_SRD_pdf.pdf?__blob=publicationFile&v=3/, 2014. [Online; Zugang: 30.01.2014].
- [Bun10] BUNDESREGIERUNG: *Energiekonzept für eine umweltschonende, zuverlässige und bezahlbare Energieversorgung*. http://www.bundesregierung.de/ContentArchiv/DE/Archiv17/_Anlagen/2012/02/energiekonzept-final.pdf?__blob=publicationFile&v=5, September 2010. [Online; Zugang: 14.04.2014].
- [Cis14] CISECO: *Datenblatt XRF Wireless UART Serial Data Module*. <http://ciseco.co.uk/downloads/documents/datasheet%20-%20XRF%20v1-6.pdf>, 2014. [Online; Zugang: 31.03.2014].
- [Dig14] DIGIKEY: *Datenblatt ZM3102N Z-Wave Module*. <http://media.digikey.com/pdf/Data%20Sheets/Zensys%20PDFs/ZM3102N.pdf>, 2014. [Online; Zugang: 31.03.2014].
- [DIN98] *Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten - Teil 11 : Anforderungen an die Gebrauchstauglichkeit - Leitsätze(ISO 9241 -11 :1998)*, 1998.

- [DIN06] *Ergonomie der Mensch-System-Interaktion - Teil 110: Grundsätze der Dialoggestaltung (ISO 9241-110:2006)*, 2006.
- [EnW14] *Energiewirtschaftsgesetz - EnWG*. http://www.gesetze-im-internet.de/bundesrecht/enwg_2005/gesamt.pdf/, 2014. [Online; Zugang: 29.03.2014].
- [EVN14] *EVN Homepage*. <https://www.evn.at/Privatkunden.aspx>, 2014. [Online; Zugang: 30.02.2014].
- [Exp14] *EXP-Tech Shop*. <http://www.exp-tech.de/>, 2014. [Online; Zugang: 30.01.2014].
- [Fal03] FALLMAN, DANIEL: *Design-oriented Human-computer Interaction*. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, Seiten 225–232, New York, NY, USA, 2003. ACM.
- [Ful65] FULLER, MCHALE: *World Design Science Decade, 1965-1975*. World Resources Inventory, 1965.
- [Hei12] HEINECKE, ANDREAS M.: *Mensch-Computer-Interaktion : Basiswissen für Entwickler und Gestalter*. X.media.pressSpringerLink : Bücher. Springer Berlin Heidelberg, 2012.
- [Hen13] HENDEN, MELDA: *Alternative Benutzerinterfaces für Smart Homes*, 2013. Bachelorarbeit.
- [Joh10] JOHNSON, JEFF: *Designing with the mind in mind : simple guide to understanding user interface design rules*. Elsevier, Amsterdam, 2010.
- [JyLY04] JIANG, LI, DA YOU LIU und BO YANG: *Smart home research*. In: *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*, Band 2, Seiten 659–663 vol.2, Aug 2004.
- [MCI14] MCI, GI FACHBEREICH: *Fachbereich Mensch-Computer-Interaktion (MCI)*, 2014.
- [Nes14] *nest Thermostat Homepage*. <https://nest.com/>, 2014. [Online; Zugang: 30.02.2014].
- [NM90] NIELSEN, JAKOB und ROLF MOLICH: *Heuristic Evaluation of User Interfaces*. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, Seiten 249–256, New York, NY, USA, 1990. ACM.
- [Nor83] NORMAN, DONALD A.: *Design Rules Based on Analyses of Human Error*. Commun. ACM, 26(4):254–258, April 1983.

- [Pic09] PICOT, ARNOLD: *E-Energy : Wandel und Chance durch das Internet der Energie*. SpringerLink : Bücher. Springer Berlin Heidelberg, 2009.
- [PTRC07] PEFFERS, KEN, TUURE TUUNANEN, MARCUS ROTHENBERGER und SAMIR CHATTERJEE: *A Design Science Research Methodology for Information Systems Research*. J. Manage. Inf. Syst., 24(3):45–77, Dezember 2007.
- [RN14] ROVING-NETWORKS: *Datenblatt RN-XV WiFly Module*. <http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Wireless/WiFi/WiFly-RN-XV-DS.pdf>, 2014. [Online; Zugang: 31.03.2014].
- [RWE14a] *RWE SmartHome Webportal*. <https://www.rwe-smarthome.de>, 2014. [Online; Zugang: 30.02.2014].
- [RWE14b] RWE: *Montage- und Bedienungsanleitung - RWE SmartHome Fernbedienung*, 2014. [Online; Zugang: 15.04.2014].
- [Sem14] SEMICONDUCTORS, NORDIC: *Datenblatt nRF24L01P+*. http://www.nordicsemi.com/kor/content/download/2726/34069/file/nRF24L01P_Product_Specification_1_0.pdf, 2014. [Online; Zugang: 31.03.2014].
- [Shn82] SHNEIDERMAN, B.: *The future of interactive systems and the emergence of direct manipulation*. Behaviour & Information Technology, 1(3):237–256, 1982.
- [Shn10] SHNEIDERMAN, BEN: *Designing the user interface : strategies for effective human-computer interaction*. Addison-Wesley, Boston [u.a.], 5th ed Auflage, 2010.
- [vRGH10] ROON, SERAFIN VON, THOMAS GOBMAIER und MALTE HUCK: *Demand Side Management in Haushalten - Methoden zur Potenzialanalyse und Kostenabschätzung*. http://www.ffe.de/download/wissen/20100406_Methodik_DSM.pdf, 2010. [Online; Zugang: 29.03.2014].
- [Zig14] ZIGBEE: *Datenblatt XBee/XBee-PRO RF Modules*. <https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf>, 2014. [Online; Zugang: 31.03.2014].